

Αντικειμενοστρεφής Προγραμματισμός (Object Oriented Programming)

Πακέτα (Packages) – Φωλιασμένες ή Εσωτερικές Κλάσεις (Inner Classes)

Παναγιώτης Σφέτσος, PhD
<http://aetos.it.teithe.gr/~sfetsos/>
sfetsos@it.teithe.gr

Περιεχόμενα Μαθήματος

- **Πακέτα (*Packages*)**
- **Φωλιασμένες ή Εσωτερικές Κλάσεις (*Inner Classes*)**

Πακέτα (*Packages*) (1/10)

- Η κύρια ιδιότητα του OOP είναι η **επαναχρησιμοποίηση κώδικα** (*code reuse*) που επιτυγχάνεται με την **κληρονομικότητα** (*επέκταση κλάσεων και διεπαφών*).
- Αυτή η μορφή επαναχρησιμοποίησης περιορίζεται στις κλάσεις του προγράμματος. ***Πως θα μπορούσαμε να χρησιμοποιήσουμε τις κλάσεις άλλων προγραμμάτων χωρίς την αντιγραφή τους ;***
- Στη Java αυτό γίνεται με την χρήση των **πακέτων**, έννοια ίδια με τις βιβλιοθήκες κλάσεων σε άλλες γλώσσες.
- Τα πακέτα **οργανώνουν συλλογές κλάσεων και διεπαφών** σε βιβλιοθήκες.

Πακέτα (*Packages*) (2/10)

- Μία άλλη χρήση των πακέτων είναι για την **αποφυγή συγκρούσεων στην ονοματολογία των κλάσεων**.
- Στις μεγάλες εφαρμογές μπορεί να δημιουργηθεί **πρόβλημα με την ονομασία των κλάσεων** (ίδια ονόματα ή ονόματα των κλάσεων της Java). Λύση στο πρόβλημα δίνουν τα πακέτα.
- Κάθε πακέτο μπορεί να περιέχει οποιοδήποτε αριθμό κλάσεων ή διεπαφών που **σχετίζονται μεταξύ τους** ως προς τον **σκοπό**, την **εμβέλεια**, ή την **κληρονομικότητα**.
- Η Java χρησιμοποιεί πακέτα για να σχηματίσει **βιβλιοθήκες κλάσεων**.
- Όλες οι κλάσεις ανήκουν σε κάποιο πακέτο, αυτό που ορίζουμε, ή το προκαθορισμένο (καθολικό).

Πακέτα (*Packages*) (3/10)

- Οι κλάσεις του κάθε πακέτου τοποθετούνται σε **ένα δικό τους κατάλογο** (*folder*). Στον κατάλογο αυτό αποθηκεύονται και τα `.class` – αρχεία.
- Όλες οι κλάσεις του πακέτου έχουν σαν πρώτη εντολή την δεσμευμένη λέξη `package`: `package <όνομα> ;`
- Τα πακέτα είναι ένας τρόπος δημιουργίας ιεραρχίας που λύνει τα προβλήματα διαχείρισης και επαναχρησιμοποίησης των κλάσεων.

Πως η Java ψάχνει και βρίσκει τα πακέτα;

- ψάχνει πρώτα τον **τρέχον κατάλογο** και μετά την **CLASSPATH** που ορίζει την διαδρομή αναζήτησης.

Πακέτα (*Packages*) (4/10)

- Οι κλάσεις ενός πακέτου περιέχουν μεθόδους όχι όμως την *main* μέθοδο.

Συνοπτικά τα πλεονεκτήματα με τα πακέτα:

- **Οργάνωση** των κλάσεων σε μονάδες
- **Επαναχρησιμοποίηση** κλάσεων άλλων προγραμμάτων
- **Αποφυγή των προβλημάτων από διενέξεις ονοματολογίας**. Διαφορετικά πακέτα μπορούν να έχουν κλάσεις με το ίδιο όνομα.
- **Ταυτοποίηση των κλάσεων** (*Java, εταιρίας, οργανισμού, κλπ.*)
- **Απόκρυψη των κλάσεων** ενός πακέτου, αν δεν θέλουμε άλλα πακέτα να έχουν πρόσβαση σε αυτές (*θα δούμε παρακάτω τις ορατότητες – πρόσβαση στα πακέτα*).

Πακέτα (*Packages*) (5/10)

- Οι κλάσεις ενός πακέτου περιέχουν μεθόδους όχι όμως την `main` μέθοδο.
- Τα πακέτα διακρίνονται:
 - ✓ στα **ενσωματωμένα** (*built in*) πακέτα της Java
 - ✓ στα **πακέτα που φτιάχνει ο χρήστης**

Built in Packages: `java.lang`, `java.util`, `java.io`, etc.

- Αυτά τα πακέτα βρίσκονται τοποθετημένα στο πακέτο “`java`”.
- Για να εισάγουμε τα πακέτα στο πρόγραμμα (είτε ενσωματωμένα, ή πακέτα του χρήστη) χρησιμοποιούμε την δεσμευμένη λέξη: **`import`**:

Σύνταξη:

```
(1) import <πακέτο>. * ;
```

Για να χρησιμοποιήσουμε όλες τις κλάσεις του πακέτου

Πακέτα (*Packages*) (6/10)

- Ο χαρακτήρας μπαλαντέρ (.*) (*wildcard*) δηλώνει όλες τις κλάσεις του πακέτου. Για παράδειγμα: **import java.io.*;**
- Προσοχή στην ιεραρχία των πακέτων – υποπακέτων και των καταλόγων που τους αντιπροσωπεύουν. Τότε η import έχει την μορφή:

```
(2) import <πακέτο>. <υποπακέτο>.* ;
```

Π.χ. **import erg1.erg2.* ;**

- Μπορούμε να κάνουμε χρήση μόνο συγκεκριμένης κλάσης ενός πακέτου:

```
(3) import <πακέτο>. <κλάση> ;
```

Π.χ. **import erg1.MyClass1;**

Πακέτα (*Packages*) (7/10)

Πακέτα που φτιάχνει ο χρήστης:

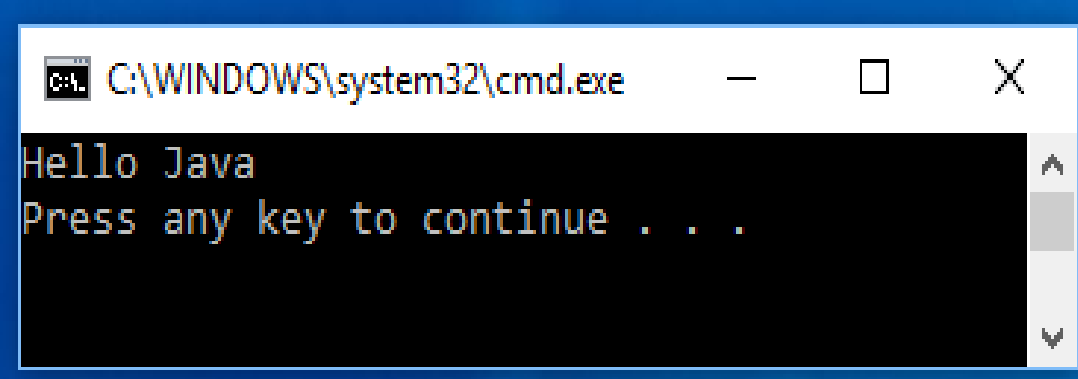
Φτιάχνουμε ένα φάκελο π.χ. `erg1`, και μέσα στο φάκελο σώζουμε την κλάση `MyClass1`.

```
package erg1    //ο φάκελος της MyClass1
public class MyClass1
{
    public void printMessage ()
    { System.out.println("Hello Java"); }
}
```

- Φτιάξαμε ένα πακέτο, που θα το χρησιμοποιήσουμε →

Πακέτα (*Packages*) (8/10)

```
import erg1.*; //όλες τις κλάσεις του πακέτου
class DemoPackage
{
    public static void main(String args[])
    {
        MyClass1 obj=new MyClass1();
        obj.sendMessage();
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the path C:\WINDOWS\system32\cmd.exe. The window content displays the output of a Java program: "Hello Java" followed by "Press any key to continue . . .". The text is displayed in a monospaced font on a black background.

Πακέτα (*Packages*) (9/10)

Επέκταση μιας κλάσης ενός πακέτου (κληρονομικότητα)

➤ Παράδειγμα:

Στο πακέτο `erg1` υπάρχει η κλάση `Employee.java`. Μια νέα κλάση η `Misthotos.java` μπορεί να επεκτείνει (κληρονομεί) την κλάση `Employee` ως εξής:

```
import erg1.Employee;  
public class Misthotos extends Employee  
{  
    // σώμα της κλάσης Misthotos  
    // κληρονομεί public και protected μέλη της Employee, όχι private  
}
```

Πακέτα (*Packages*) (10/10)

Κλάσεις με το ίδιο όνομα σε διαφορετικά πακέτα

Αν το όνομα μιας κλάσης υπάρχει σε δύο διαφορετικά πακέτα, τότε αναφορές στο όνομα της κλάσης αυτής θα πρέπει να περιέχουν και το όνομα του πακέτου. Π.χ., αν στα πακέτα `erg1` και `erg2` υπάρχει η κλάση `MyClass`, τότε:

```
import erg1.*;
import erg2.*;
public class MyDemoClass
{
    :           :
    erg1.MyClass obj1 = new erg1.MyClass();
    erg2.MyClass obj2 = new erg2.MyClass();
    :           :
}
```

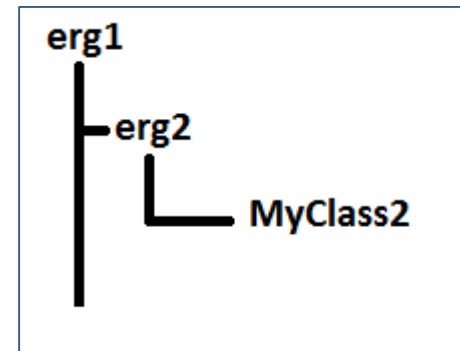
Υποπακέτα (*Subpackages*) (1/2)

- Μπορούμε να τοποθετήσουμε ένα πακέτο (2^ο) μέσα σε άλλο πακέτο (1^ο). Το πακέτο αυτό (2^ο) ονομάζεται **υποπακέτο** (*subpackage*) και βρίσκεται δεύτερο στην ιεραρχία των πακέτων. Έτσι μπορούμε να δημιουργήσουμε ιεραρχίες πακέτων.

Παράδειγμα:

Μέσα στον φάκελο **erg1** (του προηγούμενου παραδείγματος), δημιουργούμε ένα νέο φάκελο τον **erg2**, όπου αποθηκεύουμε την κλάση **MyClass2**.

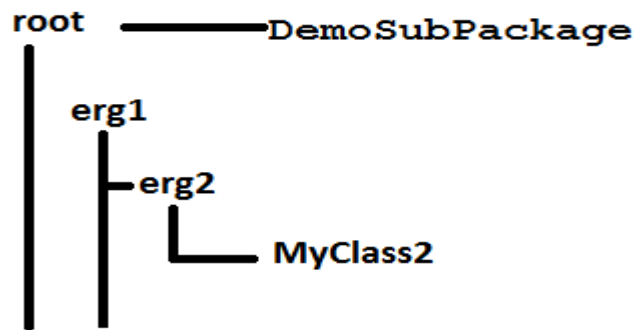
```
package erg1.erg2; //προσοχή στην ιεραρχία
public class MyClass2
{
    public void printMessage()
    { System.out.println("Hello Java"); }
```



Υποπακέτα (Subpackages) (2/2)

- Στον αρχικό φάκελο (root) αποθηκεύουμε την κλάση **DemoSubPackage**, που θα χρησιμοποιήσει την κλάση **MyClass2** του υποπακέτου **erg2**.

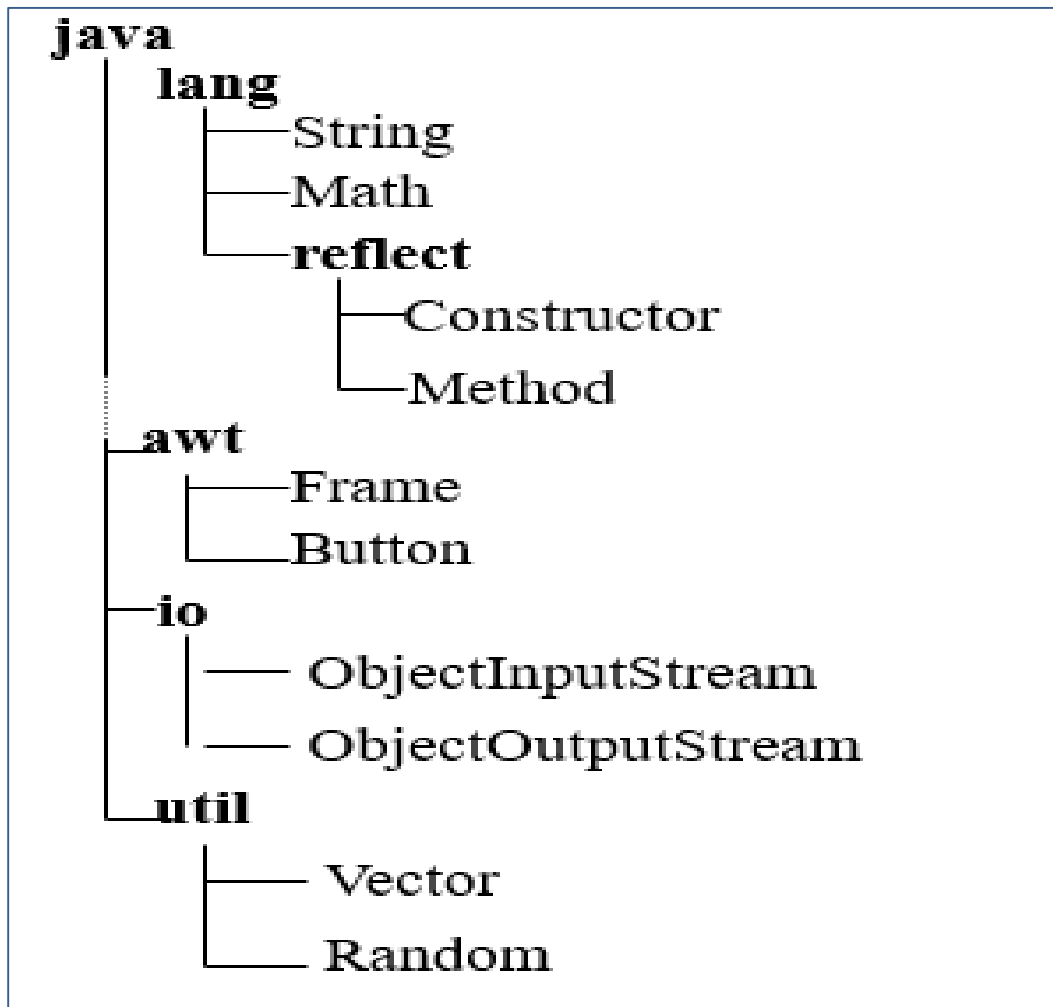
```
//προσοχή στην import
import erg1.erg2.*;
class DemoSubPackage {
    public static void main(String args[]) {
        MyClass2 obj=new MyClass2();
        obj.printMessage();}}}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window content displays the output of a Java program: 'Hello Java' followed by 'Press any key to continue . . .' on the next line. The prompt is waiting for user input.

Java built in packages

- Μερική απεικόνιση της ιεραρχίας των πακέτων της Java:



Προσβασιμότητα – Ορατότητα (1/2)

- **public:** στο όνομα της κλάσης την καθιστά προσβάσιμη/ορατή από παντού (όλες τις κλάσεις του πακέτου). Το ίδιο ισχύει και για τα πεδία και τις μεθόδους.
- **private:** πεδία ή μέθοδοι σε μία κλάση είναι προσβάσιμα/ορατά μόνο στην ίδια κλάση. Τα private μέλη δεν κληρονομούνται και δεν είναι προσβάσιμα/ορατά στις υποκλάσεις.
- **protected:** μέλη μιας κλάσης είναι προσβάσιμα/ορατά στην κλάση, υποκλάσεις και σε όλες τις κλάσεις του πακέτου.

Προσβασιμότητα – Ορατότητα (2/2)

Πρόσβαση/ Ορατότητα	public	protected	Package (default)	private
Ίδια κλάση	Ναι	Ναι	Ναι	Ναι
Άλλη κλάση στο ίδιο Πακέτο	Ναι	Ναι	Ναι	Όχι
Υποκλάση σε άλλο Πακέτο	Ναι	Ναι	Όχι	Όχι
Μη-Υποκλάση σε άλλο Πακέτο	Ναι	Όχι	Όχι	Όχι

Η μεταβλητή συστήματος CLASSPATH

Είναι η διαδρομή που η java, κατά την εκτέλεση, αναζητά τις κλάσεις και τα άλλα αναγκαία αρχεία. Ο ορισμός της διαδρομής γίνεται με δύο τρόπους:

(1) Χρησιμοποιώντας το όρισμα **-classpath** κατά την μεταγλώττιση ή εκτέλεση του προγράμματος:

```
c:> <javac/java/..> -classpath <διαδρομή1> ; <διαδρομή2>.....
```

Π.χ. C:>java -classpath C:\users\nikas\java\classes;

(2) Με την **set CLASSPATH** *(ακύρωση με την set CLASSPATH=)*:

```
set CLASSPATH=<διαδρομή1>; <διαδρομή2>...
```

Π.χ. C:> set CLASSPATH = C:\users\nikas\java\classes;

- **Φωλιασμένες ή Εσωτερικές Κλάσεις**

(Nested or Inner Classes)

Εσωτερικές Κλάσεις (*Inner Classes*)(1/13)

Java κλάσεις

- **Top level - κλάσεις**

- Ορίζονται σε ένα πακέτο
- Ορατές μέσα στο πακέτο ή και έξω από αυτό
- Ορίζονται μέσα στο δικό τους .java - αρχείο

- **Inner και nested classes - κλάσεις**

- Ορίζονται μέσα σε κλάση ή μέθοδο
- Χρησιμοποιούνται μέσα σε **εξωτερικές κλάσεις** (*outer ή enclosing κλάσεις*)
- Μπορούν να έχουν ευρύτερη ορατότητα

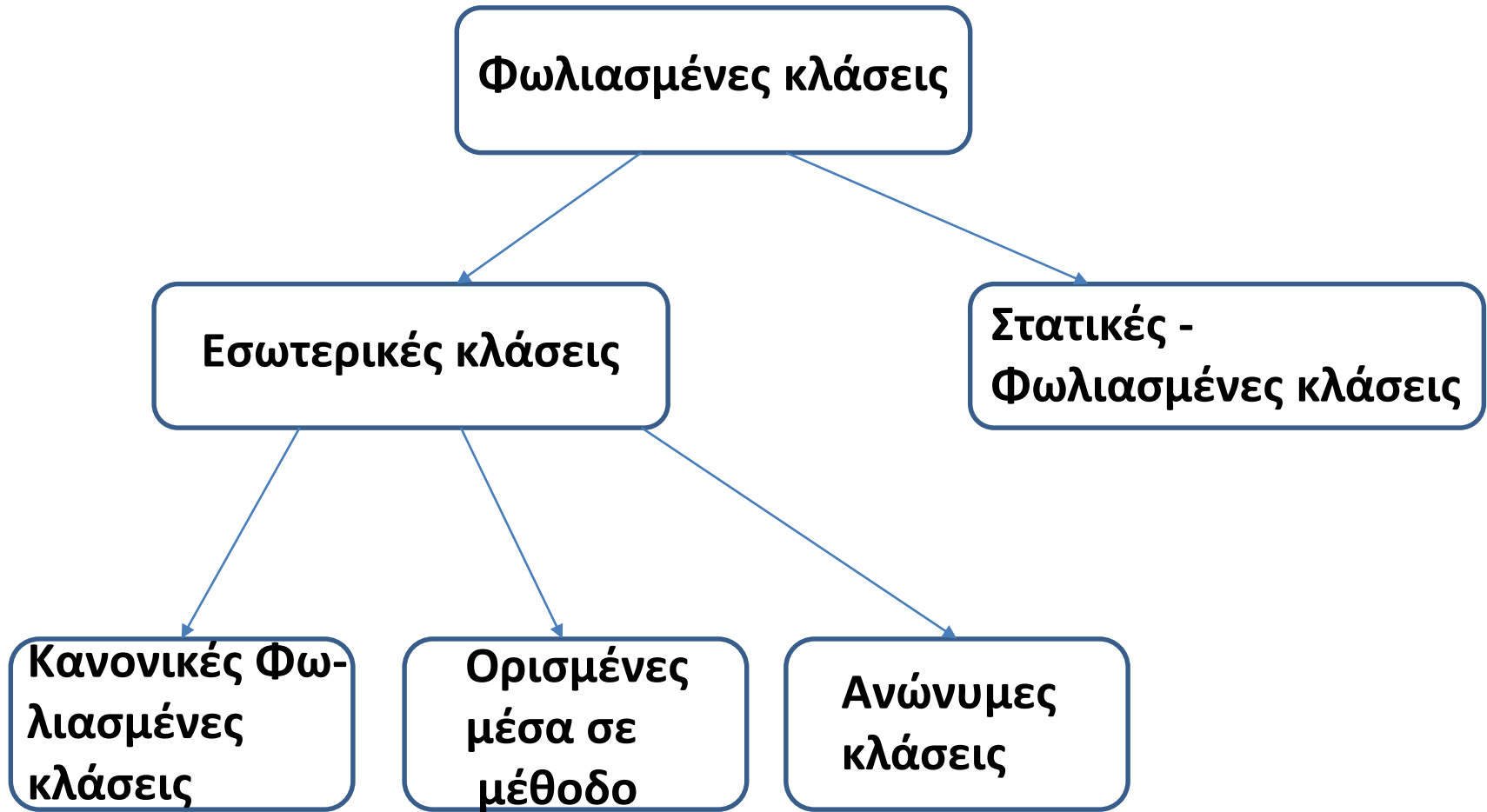
Εσωτερικές Κλάσεις (*Inner Classes*)(2/13)

- Η κλάση που ορίζεται μέσα σε άλλη κλάση λέγεται **φωλιασμένη κλάση** (*nested class*) και η κλάση που την ορίζει λέγεται **εξωτερική κλάση** (*outer class*).

Η ανάγκη δημιουργίας

- Αν και κάθε κλάση εκφράζει ένα **συγκεκριμένο σκοπό** με **ανάλογες συμπεριφορές**, ωστόσο κατά την δημιουργία μιας κλάσης υπάρχουν περιπτώσεις που ανακαλύπτουμε ότι **χρειαζόμαστε συμπεριφορά που λογικά ανήκει σε άλλη κλάση**. Η λύση σε τέτοια προβλήματα είναι η χρήση φωλιασμένων ή εσωτερικών κλάσεων. Δηλαδή, ορίζουμε μία κλάση μέσα σε άλλη.
- Οι φωλιασμένες κλάσεις χωρίζονται σε δύο τύπους:
 - Στις **μη-στατικές φωλιασμένες** (*non-static nested classes*)
 - Στις **στατικές φωλιασμένες** (*static nested classes*)

Εσωτερικές Κλάσεις (*Inner Classes*)(3/13)



Εσωτερικές Κλάσεις (*Inner Classes*)(4/13)

Εσωτερικές κλάσεις (*Inner Classes – Non-staticNestedClasses*)

- Οι εσωτερικές κλάσεις είναι ένας μηχανισμός ασφάλειας στη java. Ενώ οι κλάσεις δεν γίνεται να δηλωθούν ως **private**, ωστόσο μια εσωτερική κλάση μπορεί να δηλωθεί ως private και επίσης να γίνουν ορατά τα private μέλη της κλάσης.
- Οι **εσωτερικές κλάσεις** είναι τριών τύπων – ανάλογα με το πώς και που ορίζονται:
 - Κανονικές εσωτερικές κλάσεις (*regular*)
 - Ορισμένες μέσα σε μέθοδο (*method-local*)
 - Ανώνυμες (*anonymous*)

Εσωτερικές Κλάσεις (*Inner Classes*)(5/13)

Τύπος	Περιγραφή
Κανονικές Εσωτερικές κλάσεις	Κλάσεις που δημιουργούνται μέσα σε άλλη κλάση (όχι σε μέθοδο).
Ανώνυμες Εσωτερικές κλάσεις	Κλάση που επεκτείνουν άλλη κλάση ή υλοποιούν Διεπαφή.
Τοπικές Εσωτερικές κλάσεις Μεθόδων	Κλάσεις που δημιουργούνται μέσα σε μέθοδο.
Στατικές Φωλιασμένες κλάσεις	Στατικές κλάσεις που δημιουργούνται μέσα σε μία κλάση.
Φωλιασμένες Διεπαφές	Διεπαφές που δημιουργούνται σε κλάση ή άλλη διεπαφή.

Εσωτερικές Κλάσεις (*Inner Classes*)(6/13)

Κανονικές Εσωτερικές κλάσεις

- Δημιουργούνται απλά, γράφοντας μία κλάση μέσα σε άλλη.

```
class Outer{  
    //kodikas exoterikis klasis  
    class Inner{  
        //kodikas esoterikis klasis    }}
```

Πλεονεκτήματα:

1. Οι εσωτερικές κλάσεις έχουν πρόσβαση σε όλα τα μέλη (πεδία και μεθόδους) της εξωτερικής κλάσης, ακόμη και τα private.
2. Μορφοποιούν πιο συντηρήσιμο και ευανάγνωστο κώδικα, γιατί ομαδοποιούν τις κλάσεις και διεπαφές σε ένα μέρος.
3. Απαιτούν λιγότερο κώδικα (*βελτιστοποίηση κώδικα*)

Εσωτερικές Κλάσεις (*Inner Classes*)(7/13)

Παράδειγμα - 1ο:

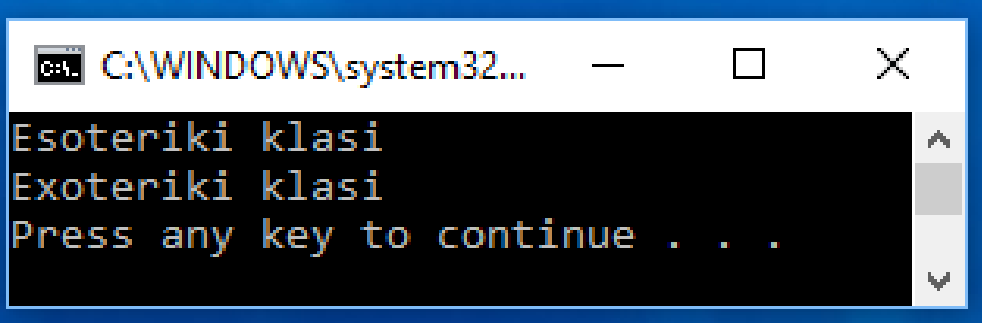
```
class Outer {
    private int metabliti;
    private String klasi = "Exoteriki klasi";

    class Inner {
        String esoterikiKlasi = "Esoteriki klasi";
        void ergasia() {
            // emfanisi tou esoterikou pediou
            System.out.println(esoterikiKlasi);

            // emfanisi tou exoterikou pediou klasi
            System.out.println(klasi); }
    }
}
```

Εσωτερικές Κλάσεις (Inner Classes)(8/13)

```
class TestIneerClass {  
    public static void main(String args[]) {  
        //exoteriko antikeimeno  
        Outer o = new Outer();  
  
        // Inner i = new Inner(); //oxi lathos  
        // Dimiourgia eidikou esoterikou-antikeimenou  
        Outer.Inner i = o.new Inner();  
        //klisi methodou tis esoterikis klasis me to i  
        i.ergasia();  
    }  
}
```

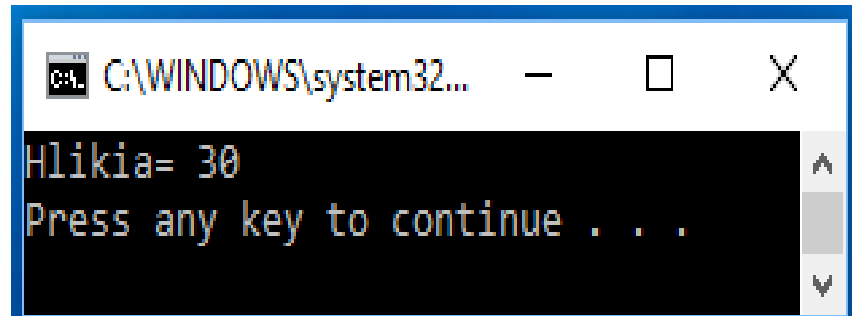


A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32...'. The window contains the following text: 'Esoteriki klasi', 'Exoteriki klasi', and 'Press any key to continue . . .'. The text is displayed in a monospaced font on a black background.

Εσωτερικές Κλάσεις (*Inner Classes*)(9/13)

Παράδειγμα - 2ο:

```
class TestOuterInner1{
    private int ilikia=30;
    class Inner{
        void ektyposi () {System.out.println ("Hlikia= "+
            ilikia) ;}
    }
    public static void main(String args[]) {
        TestOuterInner1 obj=new TestOuterInner1 ();
        TestOuterInner1.Inner in=obj.new Inner ();
        in.ektyposi ();
    }
}
```



The screenshot shows a Windows command prompt window with the title bar 'C:\WINDOWS\system32...'. The window contains the following text: 'Hlikia= 30' followed by 'Press any key to continue . . .'. The text is displayed in a monospaced font on a black background.

Εσωτερικές Κλάσεις (*Inner Classes*)(10/13)

Προσοχή:

- Αν η εσωτερική κλάση δηλωθεί **private**, τότε η κλάση δεν μπορεί να προσπελαστεί από αντικείμενο άλλης κλάσης (έξω από αυτήν). Στο παρακάτω **παράδειγμα** η πρόσβαση στην **private εσωτερική κλάση** γίνεται μέσω **public** μεθόδου της:

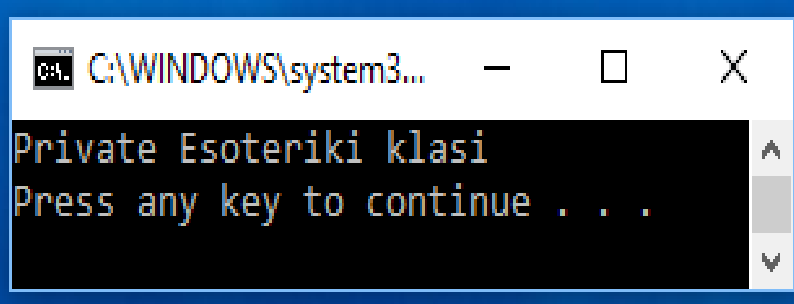
```
class Outer_1{
    int num ;
    //inner class
    private class Inner_1{
        public void emfanisi() {
            System .out.println("Private Esoteriki klasi");} }
    //Prosvasi stin esoteriki klasi meso tis methodou
    void emfanisi_Inner() {
        Inner_1 inner = new Inner_1();
        inner.emfanisi();}}}
```

Εσωτερικές Κλάσεις (*Inner Classes*)(11/13)

```
class DemoPrivateClass{
    public static void main(String args[]){

        //Dimiourgia exoterikou antikeimenou
        Outer_1 outer=new Outer_1();

        //Prosvasi meso tis methodou emfanisi_Inner
        outer.emfanisi_Inner();
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window content displays the output of the Java program: 'Private Esoteriki klasi' followed by 'Press any key to continue . . .'. The text is in a monospaced font on a black background.

Εσωτερικές Κλάσεις (*Inner Classes*)(12/13)

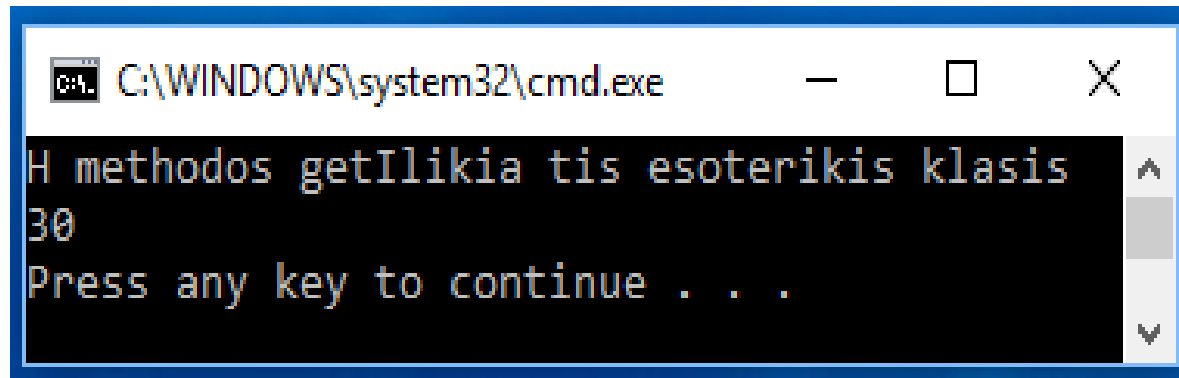
Οι εσωτερικές κλάσεις μπορούν να έχουν πρόσβαση στα **private** πεδία της εξωτερικής κλάσης, μέσω **public** μεθόδων της εσωτερικής (συνήθως τύπου `getters()`).

Παράδειγμα:

```
class Outer_2 {
    private int ilikia = 30;
    //esoteriki klasi
    class Inner_2{
        public int getIlikia () {
            System.out.println("Η methodos getIlikia tis
                                esoterikis klasis");
            return ilikia ;}
        }
    }
```

Εσωτερικές Κλάσεις (Inner Classes)(13/13)

```
class DemoPrivatePedia{  
    public static void main(String args[]){  
        //dimiourgia antikeimenou exoterikis klasis  
        Outer_2 outer=new Outer_2();  
        //dimiourgia antikeimenou esoterikis klasis  
        Outer_2.Inner_2 inner=outer.new Inner_2();  
        System.out.println(inner.getIlikia());  
    }  
}
```



```
C:\WINDOWS\system32\cmd.exe  
H methodos getIlikia tis esoterikis klasis  
30  
Press any key to continue . . .
```


Τοπικές Εσωτερικές κλάσεις Μεθόδων (1/2)

Στη Java μπορούμε να ορίσουμε μια εσωτερική κλάση τοπικά σε μια μέθοδο. Όπως και στις τοπικές μεταβλητές η εμβέλεια μιας τέτοιας εσωτερικής κλάσης, περιορίζεται στην μέθοδο. Τα αντικείμενα πρέπει να αρχικοποιηθούν μέσα στη μέθοδο.

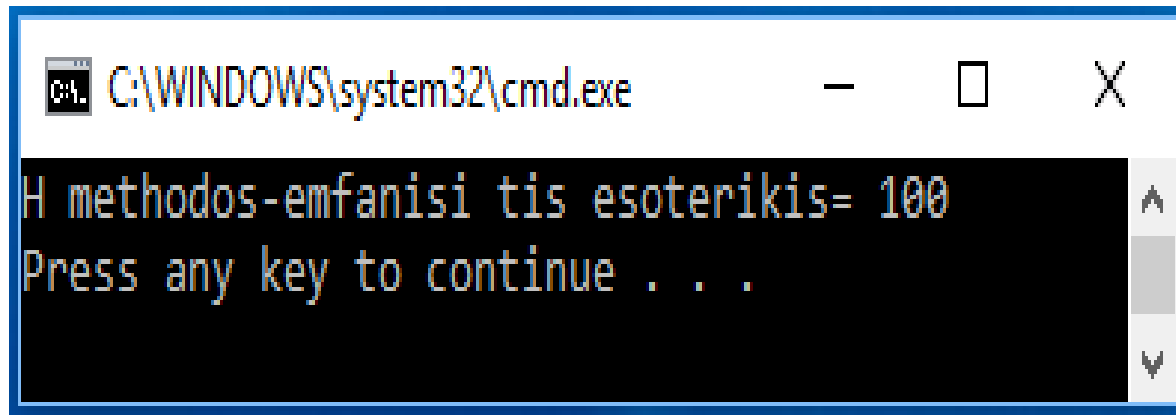
Παράδειγμα:

```
class OuterKlasi{  
    //methodos tis exoterikis klasis  
    void AMethod() {  
        int poso=100;  
        //topiki esoteriki klasi methodou  
        class EsoterikiKlasi{  
            public void emfanisi() {  
                System.out.println("H methodos-emfanisi tis  
                                    esoterikis= "+poso );}}  
    }  
}
```

Τοπικές Εσωτερικές κλάσεις Μεθόδων (2/2)

```
//Μεσα στην method προσvasi στην esoteriki klasi  
EsoterikiKlasi inner=new EsoterikiKlasi();  
inner.emfanisi();  
}
```

```
public static void main(String args[]) {  
    OuterKlasi outer=new OuterKlasi();  
    outer.AMethod(); } }
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays the output of a Java program: 'H methodos-emfanisi tis esoterikis= 100' followed by 'Press any key to continue . . .' on the next line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Ανώνυμες Εσωτερικές κλάσεις (1/2)

Μια εσωτερική κλάση χωρίς όνομα ονομάζεται **ανώνυμη εσωτερική κλάση**. Αυτές οι κλάσεις **δηλώνονται και αρχικοποιούνται ταυτόχρονα**. Συνήθως χρησιμοποιούνται όταν θέλουμε να υπερβούμε την μέθοδο μιας κλάσης ή διεπαφής.

Η σύνταξη μιας Ανώνυμης εσωτερικής κλάσης είναι:

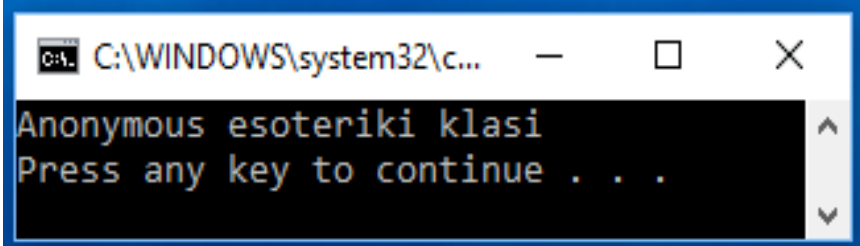
```
AnonymousInner an_inner = new AnonymousInner() {  
    public void Amethod() {  
        ...  
        ...  
    }  
};
```

Ανώνυμες Εσωτερικές κλάσεις (2/2)

Παράδειγμα: Υπέρβαση μεθόδου μιας κλάσης με την χρήση Ανώνυμης εσωτερικής κλάσης.

```
abstract class AnonymousInner{
    public abstract void mymethod();}

class Outer_Klasi {
    public static void main(String args[]){
        AnonymousInner inner= new AnonymousInner(){
            public void mymethod(){
                System.out.println("Anonymous esoteriki
                                   klasi"); }
        };
        inner.mymethod();
    }
}
```



```
C:\WINDOWS\system32\c... - □ ×
Anonymous esoteriki klasi
Press any key to continue . . .
```

Στατικές Φωλιασμένες κλάσεις (1/3)

- Ορίζονται σαν μέλη κλάσης με τον τελεστή **static**.
- Μπορούν να έχουν οποιαδήποτε ορατότητα, όπως κάθε μέλος.
- Δεν μπορούν να έχουν πρόσβαση σε μη-στατικά μέλη της εξωτερικής κλάσης.
- Εξάρτηση από την εξωτερική κλάση και όχι αντικείμενο της:

```
Outer.Nested i = new Outer.Nested();
```

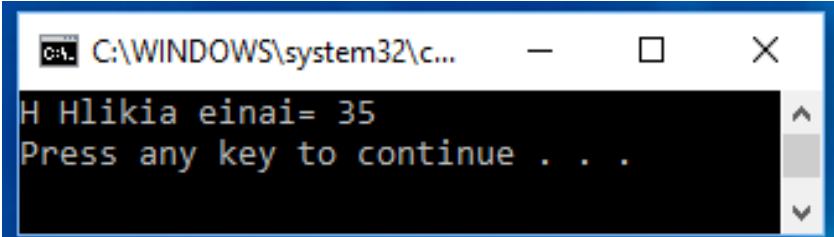
```
class Outer{  
    . . .  
    static class Nested {  
        . . .  
    }  
}
```

Στατικές Φωλιασμένες κλάσεις (2/3)

Πρόσβαση σε static μέλη της outer κλάσης.

Παράδειγμα:

```
class TestOuter{
    static int ilikia=35;
    static class Inner{
        void emfanisi () {System.out.println("Η Ηλικία
            einai= "+ilikia);}
    }
    public static void main(String args[]){
        TestOuter.Inner obj=new TestOuter.Inner();
        obj.emfanisi();
    }
}
```



The screenshot shows a Windows command prompt window with the following text:

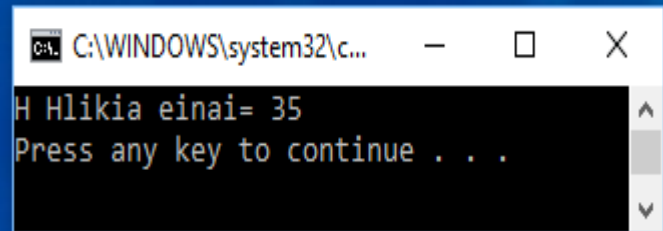
```
C:\WINDOWS\system32\c...
H Ηλικία einai= 35
Press any key to continue . . .
```

Στατικές Φωλιασμένες κλάσεις (3/3)

Πρόσβαση σε `static` μέλη της `outer` κλάσης με στατική μέθοδο. Δεν χρειάζεται η δημιουργία αντικειμένου της στατικής εσωτερικής κλάσης.

Παράδειγμα:

```
class TestOuter1{
    static int ilikia=35;
    static class Inner{
        static void emfanisi () {System.out.println("H
            Hlikia einai= "+ilikia);}
    }
    public static void main(String args[]) {
        //den xreiazetai antikeimeno tis static Inner
        TestOuter1.Inner.emfanisi ();
    }
}
```



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\c...'. The window contains the following text: 'H Hlikia einai= 35' followed by 'Press any key to continue . . .'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.